Amendments to the Claims:

1.   (Currently Amended) An apparatus comprising a computing device, wherein the computing device is configured, via a program, ~~programmed~~ to:

~~with a client that can~~ operate a client, wherein the client is configured to operate with a parser or generator for both text and binary mark up languages~~; in which~~ and the client is configured to use ~~uses~~ a unique integer value that ~~can be~~is interpreted in an index of elements, attributes and attribute values ~~needed to describe~~for describing a particular type of mark-up document,

wherein the computing device is further programmed to use the index ~~mapping~~ to map the ~~that~~ unique integer value ~~(a)~~to a token and a string;

wherein the ~~a~~token is associated with a predefined ~~prededefined~~ element, attribute or attribute value to enable a token based mark up language to be handled; and ~~also (b) to~~

wherein the ~~a~~string is associated with a ~~prededefined~~predefined element, attribute or attribute value to enable a string based mark up language to be handled.

2.   (Currently Amended) The ~~device~~apparatus of Claim 1 in which the text mark up language is XML and the binary mark up language is WBXML.

3.   (Currently Amended) The ~~device~~apparatus of Claim 1 in which a table of mappings of each of the tokens to each of the strings is created and each mapping is given one of the unique integer values.

4.   (Currently Amended) The ~~device~~ apparatus of Claim 3 in which two lists of

unique integer values are created: one indexed on tokens and the other indexed on the index of

the position of a string in a string pool table.

5.   (Currently Amended) The ~~device~~ apparatus of Claim 1 further comprising ~~in~~

~~which there is~~ an extensible framework that accepts one or more mark-up language parsers

~~and/~~or generators, each implemented as plug-ins to the framework, with different plug-ins

enabling different kinds of mark up languages to be handled by the computing device.

6.   (Currently Amended) The ~~device~~ apparatus of Claim 5 in which ~~there is a~~

~~namespace plug-in to~~ the extensible framework has a namespace plug-in that sets-up ~~all~~ the

elements, attributes and attribute values for a namespace.

7.   (Currently Amended) The ~~device~~ apparatus of Claim 6 in which the index is

encapsulated in the namespace plug-in and ~~therefore~~ is insulated from the client, parser and

generator.

8.   (Currently Amended) A method ~~of parsing a mark-up language document,~~

comprising: ~~the step of a client using~~

interpreting, by a client of a computing device, a unique integer value ~~that is interpreted~~

in an index of elements, attributes and attribute values for describing~~needed to describe~~ a

particular type of mark-up document, <u>wherein</u> the index <u>maps</u> ~~mapping that~~ <u>the</u> unique integer

value <u>to a token and a string;</u>~~:~~

     wherein the ~~(a) to~~ <u>a</u> token <u>is</u> associated with <u>a predefined</u> ~~prededefined~~ element, attribute

or attribute value to enable a token based mark up language to be handled<u>;</u> and ~~also (b) to~~

     wherein the <u>a</u> string <u>is</u> associated with a ~~prededefined~~ <u>predefined</u> element, attribute or

attribute value to enable a string based mark up language to be handled.

9.   (Currently Amended) A method ~~of generating a mark-up language document,~~

comprising<u>:</u> ~~the step of a client using~~

     <u>interpreting, by a client of a computing device,</u> a unique integer value ~~that is interpreted~~

in an index of elements, attributes and attribute values <u>for describing</u>~~needed to describe~~ a

particular type of mark-up document <u>for parsing or generating a mark up language document,</u>

<u>wherein</u> the index <u>maps</u> ~~mapping that~~ <u>the</u> unique integer value <u>to a token and a string;</u>~~(a) to~~

     wherein the <u>a</u> token <u>is</u> associated with <u>a predefined</u> ~~prededefined~~ element, attribute or

attribute value to enable a token based mark up language to be handled<u>;</u> and ~~also (b) to~~

     wherein the <u>a</u> string <u>is</u> associated with <u>a predefined</u> ~~prededefined~~ element, attribute or

attribute value to enable a string based mark up language to be ~~handled~~<u>parsed or generated</u>.

10.   (Previously presented) The method of Claim 8 in which the text mark up language

is XML and the binary mark up language is WBXML.

11.   (Previously presented) The method of preceding Claim 8 in which a table of mappings of each of the tokens to each of the strings is created and each mapping is given one of the unique integer values.

12.   (Previously presented) The method of preceding Claim 8 in which two lists of unique integer values are created: one indexed on tokens and the other indexed on the index of the position of a string in a string table.

13.   (Currently Amended) The method of preceding claim 8 further comprising accepting, in which there is an extensible framework, that accepts one or more mark-up language parsers and/or generators, each implemented as plug-ins to the framework, with different plug-ins enabling different kinds of mark up languages to be handled by the device.

14.   (Currently Amended) The method of Claim 13 in which there is a namespace plug-in to the extensible framework has a namespace plug-in that sets-up all the elements, attributes and attribute values for a namespace.

15.   (Currently Amended) The method of Claim 14 in which the index is encapsulated in the namespace plug-in and therefore is insulated from the client, parser and generator.